



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/633,197	08/07/2000	Christopher W.B. Goode	DIVA 264	3493

56015 7590 12/20/2006  
PATTERSON & SHERIDAN, LLP/  
SEDNA PATENT SERVICES, LLC  
595 SHREWSBURY AVENUE  
SUITE 100  
SHREWSBURY, NJ 07702

EXAMINER
----------

HUYNH, SON P

ART UNIT	PAPER NUMBER
----------	--------------

2623

MAIL DATE	DELIVERY MODE
-----------	---------------

12/20/2006

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

**Advisory Action  
Before the Filing of an Appeal Brief**

Application No.

09/633,197

Applicant(s)

GOODE, CHRISTOPHER W.B.

Examiner

Son P. Huynh

Art Unit

2623

**--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

THE REPLY FILED 20 November 2006 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

- a) ☐ The period for reply expires \_\_\_\_\_ months from the mailing date of the final rejection.  
b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**NOTICE OF APPEAL**

2. ☐ The Notice of Appeal was filed on \_\_\_\_\_. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

**AMENDMENTS**

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because  
(a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);  
(b) ☐ They raise the issue of new matter (see NOTE below);  
(c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or  
(d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: \_\_\_\_\_. (See 37 CFR 1.116 and 41.33(a)).

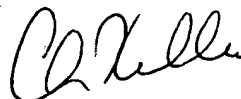
4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).  
5. ☐ Applicant's reply has overcome the following rejection(s): \_\_\_\_\_.  
6. ☐ Newly proposed or amended claim(s) \_\_\_\_\_ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).  
7. ☒ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.  
The status of the claim(s) is (or will be) as follows:  
Claim(s) allowed: \_\_\_\_\_.  
Claim(s) objected to: \_\_\_\_\_.  
Claim(s) rejected: 1-6, 8-19, 22 and 23.  
Claim(s) withdrawn from consideration: \_\_\_\_\_.

**AFFIDAVIT OR OTHER EVIDENCE**

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).  
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing of good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).  
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

**REQUEST FOR RECONSIDERATION/OTHER**

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because:  
See Continuation Sheet.  
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08) Paper No(s). \_\_\_\_\_.  
13. ☐ Other: \_\_\_\_\_.



**CHRIS KELLEY  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2600**

Continuation of 11. does NOT place the application in condition for allowance because: In response to Applicant's request to the Examiner to consider the reference in Information Disclosure Statement received on January 15, 2002, the Examiner provides a copy of considered listed documents.

In response to Applicant's arguments that certain subject matter in Huston does not have full support in the provisional application 60/176,666 (page 7, paragraph 2), and Huston does not teach or suggest providing usage statistics to the content provider (page 9, paragraph 4, page 10, paragraph 5-page 11, paragraph 4), the Examiner respectfully disagrees.

The Applicant does not specifically provide what subject matter is not full supported in Huston's provisional application. The Examiner relies on Huston for the teaching of providing statistic to content provider, and resource lease. In fact, Huston's provisional application (60/176,666) discloses Content Exchange is enabled when a content provider contracts with an access provider. In return, the access provider makes a certain number of caches available, and allows the content provider to control when and what content should be server. The access provider also provides information on cache access rates so the content provider can track how often the content is accessed. The amount of data severed is the basis for the billing arrangement performance between the content and access provider (page 3, paragraph 5); Content Exchange enables a new business model of hosters and access providers because they are enable to guarantee the availability of certain content in a network of proxy caches. Depending on the agreement with the content provider, new content is guaranteed to be available as soon as the content provider delivers it, and to remain in the proxy caches as long as the content provider specifies. In essence, the content provider rents spaces in the proxy caches in order to guarantee faster performance and higher availability for their content compared to content providers who are not using Content Exchange; Billing can be done using a variety of methods, including number of bytes served, amount of storage reserved and used, duration of the storage, and transfer rate. Content Exchange also enables another revenue stream by providing access statistics to the content provider. By collecting cache hit information, the hoster or access provider can provide an accurate picture of which data is being access and how often. Click through rates as well as patterns of click through can be generated (see included, but is not limited to, page 4, paragraphs 1-3, page 12, paragraphs 1, 3, 6, page 14, paragraph 4-page 15, paragraph 2). Thus, the limitations of "providing statistic to at least one content provider" (interpreted as providing access statistics, amount of storage reserved and used, transfer rate, duration of storage, etc. to the content provider) and "resource lease" (interpreted as content server contracts with an access provider, content provider rents space in the proxy caches of hoster/access provider, and access provider makes/guarantee certain of caches available according to contract/agreement with the content provider) ; the limitation "providing usage statistics to the content provider" is interpreted as the access server, hoster, provides access statistics, amount of storage used, transfer rate, cache hit information, or which data is being accessed and how often, etc. to the content provider.

Therefore, the provisional application of Huston fully supports the subject matter that Examiner relies on in the rejection.

Applicant further argues Sie fails to teach "a server complex, at a cable television system operator location, comprising a plurality of partitions, each of said partitions storing video assets provided by respective content supplier" because Applicant argues there is nothing in Sie regarding partitioning of subscriber server so that each partition is used for storing video assets associated with one content supplier; the server, along with the program request database 136, are part of the system of an additional content provider; the program server belongs to a particular provider of additional content, and not part of a server complex at a cable television system operator location (see page 8, paragraph 2-page 9, line 2). These arguments are respectfully traversed.

Neither limitation "partitioning subscriber server so that each partition is used for storing video assets associated with one content supplier" nor "program server is part of a server complex at a cable television system operator location" is recited in the claims. Claim 13 recites "a server complex, at a cable television system operator location, comprising a plurality of partitions, each of said partitions storing video assets provided by respective content supplier" (lines 3-5). Sie discloses program server comprises a plurality of buffers and/or platters and corresponding write heads, each write head could write a different digital channel on its respective surface of the platter and simultaneously store multiple digital channels (see include, but not limited to, col. 19, line 16-col. 21, line 6), wherein the digital channels are provided by the content providers such as provider of commercial supported channels, provider of commercial free channels, providers of video on demand channel, etc. - see include, but is not limited to, lines 20-36, figures 1-3, 14-16); cable television provider/program delivery system interacts with the program server and/or systems of additional content providers to control the operations of the system (see include, but is not limited to, figure 1, col. 3, line 56-col. 4, line 67). Thus, the limitation of "server complex at a cable television system operator location, comprising a plurality of partitions, each of said partitions storing video assets by respective content supplier" is broadly interpreted as program server with/without other parts of delivery system/cable television provider at delivery system/cable television provider, comprising a plurality of buffers/platters/disk drive, each buffer/platter/disk drive stores video data in each digital channel provided by respective content providers (e.g. provider for video on demand channel, provider for commercial free channel, etc.) .

In addition to application argument that Sie discloses the program server is used for storing programs associated with an additional content provider (e.g. col. 4, lines 63-64). This server, along with the program request database 136, are part of the system of an additional content provider (e.g. Sie, col. 3, lines 60-63) and then the applicant concludes the program server of Sie belongs to a particular provider of additional content, and not part of a server complex at a cable television system operator location (see page 8, paragraph 4), the Examiner respectfully disagrees. The Examiner agrees with Applicant's argument that Sie discloses the program server stores programs associated with an additional content provider; the program server and program request database are part of a system of an additional content provider. However, There is nowhere in the Sie discloses that the program server of Sie is not part of a server complex at a cable television operation location as concluded by the Applicant. The disclosure of "part of" or "stores program associated with..." in Sie's disclosure could be part/portion at the cable service provider/delivery system that is allocated/assigned for content provider to store its content.

Therefore, Sie's disclosure is met the limitation "a server complex, at a cable television system operator location, comprising a plurality of partitions, each of said partitions storing video assets provided by respective content suppliers;"

In response to applicant's argument that there is no suggestion/motivation to combine the references (page 9, paragraph 5-page 10, paragraph 4), the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the motivation is found in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

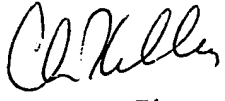
In specifically, Sie discloses a system comprises service provider for receiving content from content source, storing content, and providing the content to the user, and collecting statistics/ usage information (see include, but is not limited to, figure 1, col. 3, line 56-col. 4, line 6, col. 5, lines 50-60, col. 19, line 15-col. 20, line 34).

Gordon also discloses a system comprises service provider for receiving content from content source, storing content, and providing the content to the user, and collecting statistics/ usage information (see include, but is not limited to, figures 1-5, col. 4, line 28-col. 5, line 13, col. 7, line 26-col. 8, line 60). Gordon additionally discloses increasing and decreasing the number of copies of any video asset in response to usage statistics (see including, but is not limited to, col. 8, line 40-col. 9, line 13) is interpreted as the claimed limitation increasing and decreasing a capability of the memory resource in response to the usage statistic. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sie to use the teaching as taught by Gordon in order to improve efficiency in data transmission such as minimizing video asset data blocked down during peak time and reduce delay time to provide requested data to subscriber (col. 2, line 60-col. 3, line 10).

Huston also discloses a system comprises service provider for receiving content from content source, storing content, and providing the content to the user, and collecting statistics/ usage information (see include, but is not limited to, figures 2a, paragraphs 0020-0024, 0037, 0041, 0056, 0063-0064 of 2002/0007402 A1 and/or see the provisional application, page 3, paragraphs 5, page 4, paragraphs 1-2, page 9, paragraph 2, page 12, paragraphs 1, 3, 6, page 14, paragraph 5-6, page 15, paragraph 2). Huston additionally discloses providing statistic to content provider and resource lease (interpreted as hoster/access provider provides statistic information, cache access rate, amount of storage used, etc. to the content server and the content server rent space in proxy server of hoster/access provider, or contracts with an access provider for number of cache available (see including, but is not limited to, paragraphs 0064-0072, 0063, 0066 of US2002/0007402 A1 and/or see the provisional application, page 3, paragraphs 5, page 4, paragraphs 1-2, page 9, paragraph 2, page 12, paragraphs 1, 3, 6, page 14, paragraph 5-6, page 15, paragraph 2). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sie in view of Gordon to use the teaching as taught by Huston in order to allow content provider to better manage their content and/or guarantee faster performance and higher availability (see paragraph 0016, lines 8-11 of US 2002/0007402 A1 and/or see the provisional application, page 1, Introduction, page 4, paragraphs 1-2).

Therefore, the combination of the references is properly applied.

For the reasons given above, rejections on claims 1-6, 8-19, 22, and 23 are maintained as discussed in the Final Office Action dated 09/26/2006.

  
CHRIS KELLEY  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2600

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US01/24624

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H04N 9/00, 7/16, 7/173

US CL : 725/9, 86, 87, 95, 93, 91, 92

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 725/9, 86, 87, 95, 93, 91, 92

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
NONEElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
EAST - bandwidth, allocation, service provider, subscriber, compensation

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,673,430 A (STORY) 30 September 1997, whole document	1-21
A	US 5,742,892 A (CHADDHA) 21 April 1998, whole document	1-21
A	US 5,732,216 A (LOGAN et al) 24 March 1998, whole document	1-21

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G* document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means	
*P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

09 SEPTEMBER 2001

Date of mailing of the international search report

10 OCT 2001

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

VIVEK SRIVASTAVA

Telephone No. (703) 305-4038

50269-038

Patent

UNITED STATES PROVISIONAL PATENT APPLICATION

FOR

CONTENT EXCHANGE

INVENTORS:

ARTHUR CHARLES THOMAS HUSTON  
MATTHEW DENNIS HAINES  
FREELAND KNIGHT ABBOTT, III  
ADAM LOUIS BEGUELIN  
MICHAEL ALEXANDER CHOWLA

PREPARED BY:

MCDERMOTT, WILL & EMERY  
600 13<sup>TH</sup> STREET, N.W.  
WASHINGTON, DC 20005-3096  
(202) 756-8000

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL402671424US

Date of Deposit January 18, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

CLARE C. FINNEY

(Typed or printed name of person mailing paper or fee)

Clare C. Finney  
(Signature of person mailing paper or fee)

50269-038

Inktomi Corporation.

# Content Exchange

*Concepts and Inventions*

50175555-01600

## Table of Contents

INTRODUCTION .....	3
NEW BUSINESS MODEL .....	3
CACHE COHERENCE AND VALIDATION.....	4
VARYING ON USER AGENT .....	5
RETRYING FAILED REQUESTS .....	7
UPDATING A NETWORK OF PROXY CACHES.....	7
CACHE PREFETCH.....	8
CACHE FAIL-OVER .....	10
PROVISIONING FOR CACHE OBJECT TYPES AND QUOTAS .....	11
FEEDBACK ON NEW CONTENT .....	12
DISTRIBUTED LOG ENTRY AGGREGATION .....	12
BILLING AND STATISTICS COLLECTION .....	14

003110-9992109



# Content Exchange 1.0

## *Concepts and Inventions*

### Introduction

Content Exchange allows content providers to gain greater control over the placement of content in the Internet, resulting in the delivery of up-to-date content, increased performance, and increased feedback into access of content. This is accomplished by explicitly manipulating content in caches at an access provider instead of merely updating origin servers, resulting in the following benefits:

- Updating content is predictable and fast because stale data can be deleted and new data updated under the control of the content provider instead of using cache heuristics and HTTP protocols
- Users experience increased performance because content is immediately available in the access provider's caches instead of requiring trips to the origin server(s)
- Content providers gain increased visibility into the access of their content in the cache

Content Exchange is enabled when a content provider contracts with an access provider. In return, the access provider makes a certain number of caches available, and allows the content provider to control when and what content should be served. The access provider also provides information on cache access rates so the content provider can track how often the content is accessed. The amount of data served is the basis for the billing arrangement performance between the content and access provider.

In certain situations, a hoster may also be involved in the Content Exchange. In this case, the content provider has control over one or more caches at the hoster, and it is the hoster's responsibility to forward the content to one or more caches at the access provider.

### New business model

#### Standard model

Standard proxy caches make no guarantee as to which content will be stored in the cache, and for how long. Storing and removing data from the cache is determined by when and how often it is requested, and by heuristics about object and cache size, as well as by checking the origin server to see when objects become stale. Because all data is treated the same, the only way a hoster or access provider makes money from proxy caches is by making the network more efficient without adding costly hardware, and perhaps by attracting and keeping customers.

## Content Exchange

Content Exchange enables a new business model for hosters and access providers because they are able to guarantee the availability of certain content in a network of proxy caches. Depending on the agreement with the content provider, new content is guaranteed to be available as soon as the content provider delivers it, and to remain in the proxy caches as long as the content provider specifies. In essence, the content provider rents space in the proxy caches in order to guarantee faster performance and higher availability for their content compared to content providers who are not using Content Exchange.

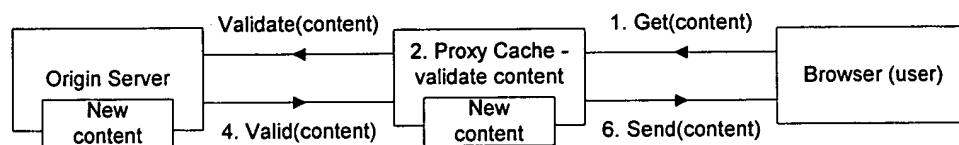
Billing can be done using a variety of methods, including number of bytes served, amount of storage reserved and used, duration of the storage, and transfer rate. Content Exchange also enables another revenue stream by providing access statistics to the content provider. By collecting cache hit information, the hoster or access provider can provide an accurate picture of which data is being accessed and how often. Click-through rates as well as patterns of click throughs can be generated.

The remainder of the document describes the technologies used to provide these benefits and to support the new business and billing models. More details on billing and statistics collection are in the [Billing and Statistics collection](#) section in the back of this document.

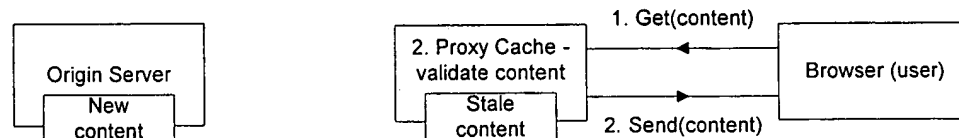
## Cache coherence and validation

### Standard model

Web proxy caches have traditionally been a loosely coupled set of caches with no validation support from the origin servers. As a result, web proxy caches have developed a unique set of heuristics for maintaining cache coherence. However, without direct validation support from the origin servers, these techniques are overly conservative, resulting in unnecessary trips to the origin servers and a degradation in cache performance. Despite this, in certain cases the algorithms still permit the possibility of inconsistent caches.



Conservative heuristic: Unnecessary trip to the origin server, resulting in delays



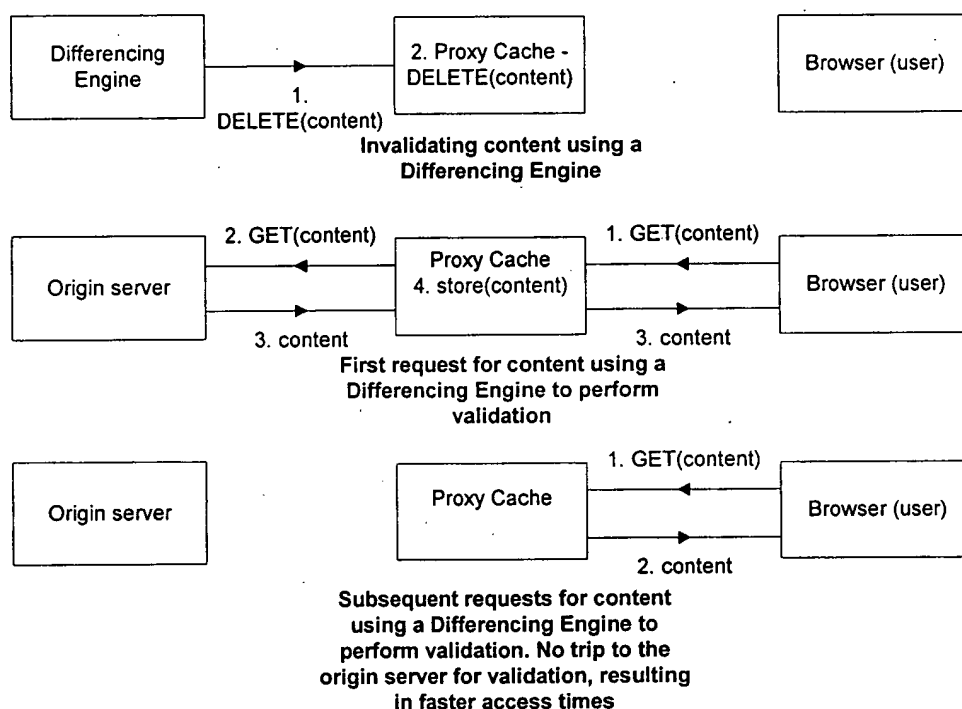
Optimistic heuristic: No trip to the origin server, resulting in stale content

## Content Exchange

The Content Exchange model provides greater control over cache coherence and validation by providing a mechanism to explicitly update and invalidate content. The result is that we can place content into caches and ignore all previously-used web proxy caching heuristics for

determining the validity of the content, and instead rely on an accurate validation model. This concept is described in a patent application already submitted by Webspective (now part of Inktomi). It is described here because of its relevance to Content Exchange, and because Content Exchange provides certain enhancements.

Content Exchange validation uses a "differencing engine" to compare old content to new content and to generate a set of deltas for objects that have been added, changed, or deleted. In the Content Exchange implementation, this is performed using the Binary Content Distributor (BCD) of Inktomi's Content Distribution Suite; it could, however, be any differencing engine. The differencing engine sends an HTTP DELETE to one or more proxy caches for each object that has been added or changed. (The BCD also sends a DELETE for each object that has been added; this is used to implement cache prefetch, described below). The proxy cache is configured to not perform validation for these objects, instead relying on notifications from the differencing engine to tell it when objects become invalid.



## Varying on User Agent

### Standard proxy cache

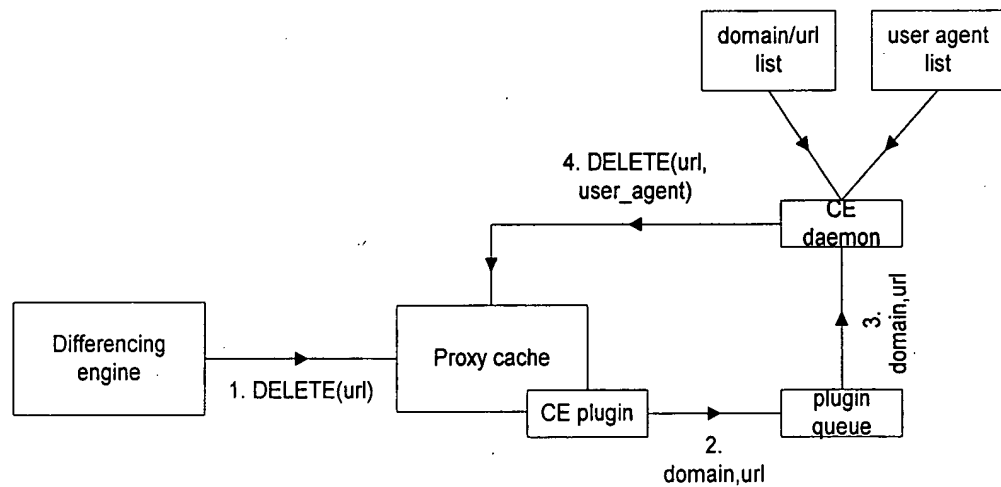
Web proxy caches traditionally store cached content indexed by URL. This can be insufficient, however, if the content varies on the user agent. The user agent is a field in the HTTP header that indicates something about the client that is making the request. Usually, it specifies the type of the browser. If browsers (for example Internet Explorer and Netscape) and/or browser versions offer different capabilities, the origin server can serve different content depending on which browser served the request. Because of this, an exact match on URL in a proxy cache is insufficient; what is required is a match on the combination of URL and user agent.

Inktomi's Traffic Server offers the capability of varying on user agent by storing objects in the cache using a hash table of URL and user agent. Only an exact match on both will serve the content; if the match is not exact the request will be forwarded to another proxy server or the origin server, depending on how the Traffic Server is configured. What this means, however, is that a DELETE request from the differencing engine (BCD) will not match on any of the URL/user-agent combinations in the Traffic Server.

### Content Exchange

The ideal solution to the problem of deleting an object for all user-agents would be to store objects indexed on URL and within URL by user-agent, enabling a two-phase search to take place, or an iteration through all the user-agents. This could be accomplished in a variety of ways, including a binary tree (btree) by URL and user-agent. It would be simple to search for a URL/user-agent pair, or to search for a URL and empty user-agent and walk through the tree until the user agent changed. Another approach would be to have separate indices for URL and user-agent within URL, either using btrees or hash tables.

The solution used in Content Exchange is to capture the DELETES from the differencing engine in a plugin and re-issue it once for each of a list of frequently requested user-agents. In this approach, a list of domains or content which are Content Exchange-enabled is used to determine which DELETES should receive the special processing (it is also possible to do it for all DELETE requests). If the domain appears in the list, the plugin places the request in a queue, where it is processed by a daemon process which sends a DELETE to the Traffic Server for each user-agent. (It would also be possible to do this in the plugin). The HTTP request contains the **Max-Forwards: 0** directive so it is not forwarded to the origin server; if the URL/user-agent pair is not found, the daemon ignores the error. The list of frequently requested user-agents can be generated manually or by monitoring and tabulating browser requests.



Varying on user agent

To prevent a DELETE request for a user-agent being put back in the plugin-queue, resulting in an endless loop, the daemon adds a special field to the HTTP header indicating to the plugin that the request should not be processed. Alternatively, the plugin could check for the presence of a user-agent in the request and ignore the request if the user-agent is found.

## Retrying failed requests

### Standard proxy caches

If a request fails in a standard proxy cache, perhaps because the origin server is off-line, an error is returned and the request is not re-tried. This is all that is possible because the request is essentially stateless; the proxy queue can assume that the user making the request will try again later when the error condition has been fixed. This is insufficient for Content Exchange, however, because the integration of a differencing engine and a network of proxy servers reflects the state of new content, and must be processed correctly in order to keep the caches synchronized and updated.

### Content Exchange

Because Content Exchange places requests in a queue and processes them in a daemon, it is possible to retry requests that may fail because the Traffic Server has failed or been temporarily taken out of operation. If this occurs, the daemon logs a message and places it in a retry queue indicating the hostname of the Traffic Server, the domain and url names, an explanation of the error, number of tries and the first and last retry times. A thread in the daemon reads the retry queue and retries the event at regular intervals until the event succeeds, the domain is removed from the Content Exchange domain list, or a maximum number of retry times is exceeded. Retry may be implemented at the granularity of URL, port number on the proxy cache (the cache may have multiple port numbers), URL and user-agent, or any combination.

Because the connection between the differencing engine and Traffic Server may fail, retry can be implemented by creating a daemon which mimics a Traffic Server and forwards requests from the differencing engine to the actual Traffic Server. Any errors that may occur are placed in the retry queue. Alternatively, the differencing engine can handle retry on its own.

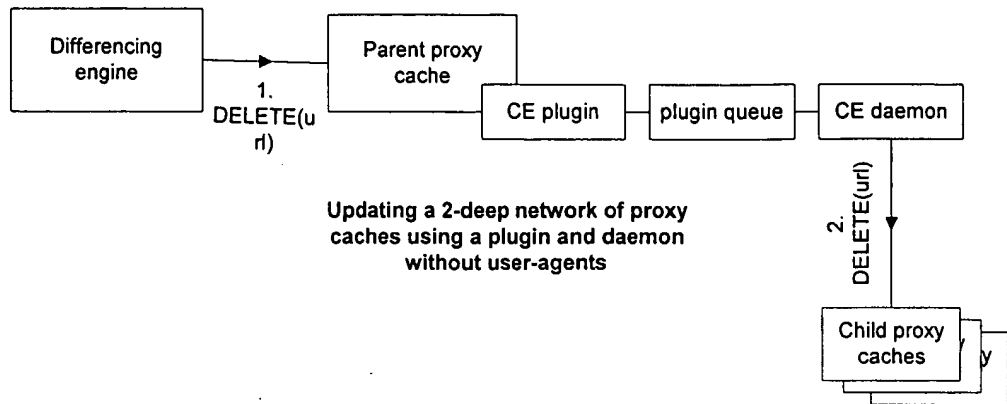
## Updating a network of proxy caches

### Standard proxy caches

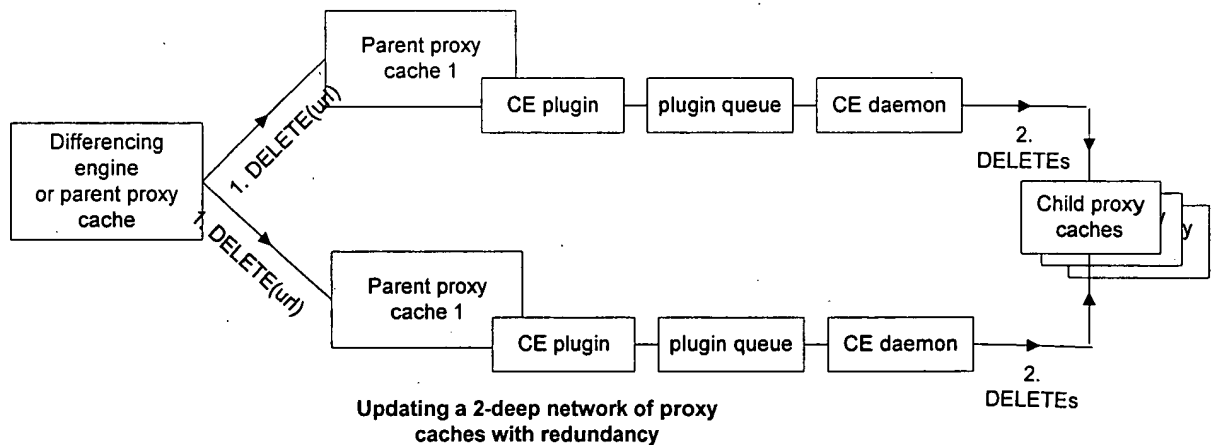
In certain situations, it may be advantageous to update an entire network of proxy caches without requiring the differencing engine to know about all the proxy caches. For example, it may take too long for the differencing engine to loop through all the proxy caches, or the differencing engine may be located outside the local area network, perhaps belonging to another organization. The latter case may occur when the differencing engine is running at a content provider; when content changes the content provider uses the engine to forward the changes to proxy cache(s) running at a hoster or access provider. In this case the system would be easier to configure and more secure if there was a single (or few) points of interaction between the content provider and hoster/access provider.

### Content Exchange

Content Exchange enables updating a network of proxy caches by putting caches into parent-child relationships, and making the parent caches responsible for updating the children. In this way, the DELETE requests are forwarded from the parent traffic servers to the children. The hierarchy can be as shallow or deep as required; if it is shallow, the parent proxy cache (actually the daemon running on the parent proxy cache) is responsible for updating all the children. If the hierarchy is deep, there may be multiple nested proxy caches. If the proxy caches are varying on user-agent, only the DELETE request for the first user-agent should be processed in the plugin. The child proxy caches can be updated using either iterative unicasts, multicast, or broadcast.



If the parent proxy cache stops operating, child caches may not receive the DELETE messages, resulting in an inconsistent network and potentially stale content. To make the system redundant, Content Exchange enables two or more proxy caches to serve as parents to the same child proxy caches. Each parent proxy cache is responsible for forwarding the same DELETE requests. In order to avoid processing the requests more than once, the children store completed requests in a queue; requests that are received but are already in the queue within a certain small time limit are ignored as redundant.



#### Security

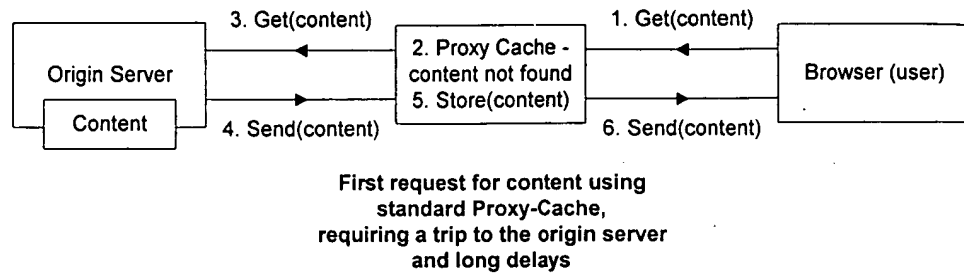
Security is provided in Content Exchange to ensure that only proxy caches within the network and trusted hosts outside the network can issue DELETES. This is accomplished by only processing requests from certain IP addresses. The listing of domains and URLs that are Content Exchange-enabled also prevents an outside party from using space in the cache for unauthorized content. Security can be further tightened by ignoring multiple redundant requests within a certain time period.

#### Cache prefetch

#### Standard model

Web proxy caches traditionally offer no guarantee of what content is available. Content is not fetched from the origin server until at least one user requests it, resulting in potentially long

delays. In addition, caches use an approximation of a least-recently-used algorithm to determine which data should be dropped when the cache becomes full. If the data is dropped, the next user to request it will again experience a long delay.



### Content Exchange

Content Exchange offers an alternative model for determining which content should be stored in a cache. Content can be fetched from the origin server and stored in the cache before a user requests it. This means that the cache will contain both objects being stored as the result of a client fetch (the traditional mode of operation for a proxy cache), as well as objects that have been pre-loaded into the cache by a content provider to ensure fast and reliable access to the content.

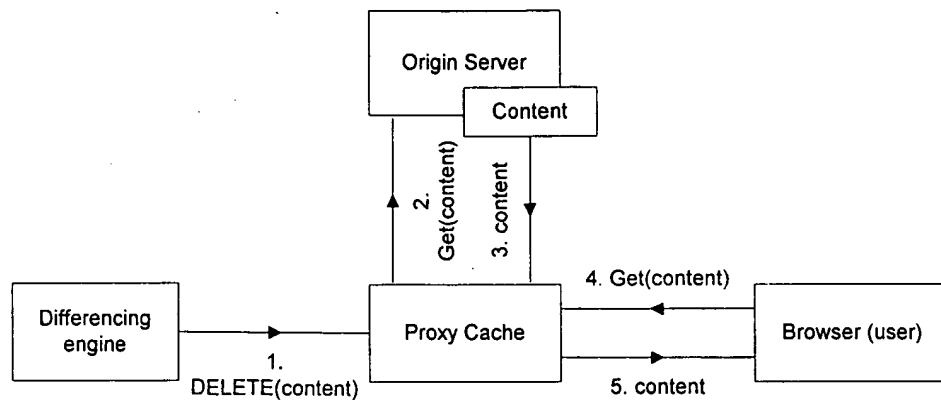
The implementation of prefetch is an extension of the validation model and again uses a differencing engine to inform a proxy cache or network of proxy caches about new content. In its simplest form, the differencing engine must send an HTTP DELETE for content that has been added, as well as for content that has been changed and deleted. The proxy cache, plugin, or daemon trap the DELETE and issue a GET back to the proxy cache. As in the discussion of validation in the previous sections, multiple user agents are handled, as well as well as queuing, retry, updating a network of caches, redundancy, and security.

If the differencing engine issues only DELETES, the proxy cache will attempt to delete content that is newly-created and cannot exist in the cache, and to GET content that has been removed from the origin server. The HTTP not found status code can be safely ignored, however, the processing of the request wastes processing time. This can be fixed by extending the differencing engine to issue different requests for loading new data, refreshing changed data, and deleting obsolete data.

It would also be possible to push new content out from the differencing engine instead of issuing requests and pulling the data from the origin server. This would save a small amount of network bandwidth and latency, however, it would leave the proxy cache open to a security attack placing erroneous data on the cache.

Content that is enabled for Content Exchange is also "pinned" in the cache so that it stays in the cache even when space is required for new content. Pinning means that other content is selected for deletion even though the Content Exchange content may be less frequently or less recently accessed. The prefetching and pinning of content provides a guaranteed availability in the cache, enabling the revenue model discussed in previous sections.

Any object which is not specified in the domain/URL list will be subject to standard proxy cache operation, e.g. the first user to access it will incur the overhead of a round-trip to the origin server and the object may be deleted if cache space must be re-used.



**Prefetching content**  
 Steps 1-3 occur during prefetch;  
 Steps 4-5 occur when content is  
 requested

Content Exchange addresses the problem of enabling fast access to content without multiple copies of the data by prefetching the content onto a small number of proxy caches and configuring the child proxy caches to check the parent proxy cache(s) for content before going to the origin server. In this way all content is processed from a local high-speed network without preloading all the caches.

The access patterns in the network can be used to generate the list of children for a parent: if the parent notices a request from a child, it can add that hostname to the list of children and forward subsequent Content Exchange updates to the child. If it later notices that it has not received such a request for a long period of time, it can then remove the child from the list. This is one approach to eliminating configuration problems in the system: adding a parent relationship to the child will automatically propagate the opposite relationship to the parent.

## Cache fail-over

### Standard model

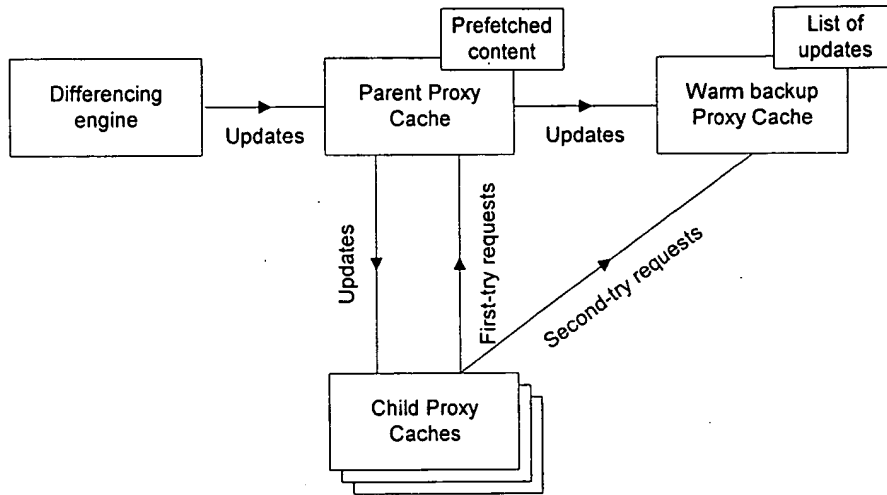
Using multiple parent proxy caches to prefetch the same content provides redundancy: if a parent proxy cache stops operating, the child proxy cache can try one of the other parent proxy caches before going to the origin server. This ensures high availability and access to content on a high-speed network, but requires redundant copies of the data, using up a lot of space on the proxy caches and network bandwidth when the caches are pre-fetched.

### Content Exchange

Content Exchange enables a proxy server to be designated as a warm backup, and as an alternative parent to the other child proxy servers. The warm backup receives the same HTTP updates as the parent proxy server (either from the parent or directly from the differencing engine) but instead of doing the prefetch, it stores the requests in a file. When a new series of requests is received, the old list is deleted and the new list is written. When the warm backup receives a request from a child (or in fact any request) it will automatically check the parent proxy cache. If that request fails, the warm backup then executes all the requests that it had previously stored, effectively prefetching all the content. The warm backup can then process all child requests through the high-speed network without going to the origin server.



Further redundancy can be achieved by assigning warm backups to other warm backups; if both the parent proxy server and first warm backup fail, the second warm backup will receive a request from another child and request it from the first warm backup. When that request fails, the second warm backup takes over.



Using a warm backup to implement redundancy

## Provisioning for Cache Object Types and quotas

### Standard model

Traditional proxy caches represent an expanse of space for storing all types of proxied objects. However, the space requirements of these objects can vary widely and, as a result, it is possible for one type of object to displace other (smaller) object types. Such is the case with streaming media objects versus HTTP objects, where one of the former objects can displace thousands of the latter.

### Content Exchange

Content Exchange provides a technique to provision the cache such that users can control the bandwidth requirements of different object types within the total cache space. At the most basic level, Content Exchange provisions object types by validating, prefetching, and pinning objects for content providers who are using Content Exchange. In this case domain-names (for example [www.company.com](http://www.company.com)) or URLs are used to distinguish object types. More granularity can be achieved by listing individual URLs or by using regular expressions (wild-carding). Further selection can be performed by specifying certain mime-types (for example graphics image (.gif) files) or minimum object sizes. Virtually any portion of the HTTP header can be used to determine which objects get prefetched and which do not.

Object provisioning can be extended by embedded directives within the HTTP requests that are sent from the differencing engine or origin server to the proxy cache(s). When sent from the differencing engine, some objects may be designated only for validation and not for prefetch. A content provider using Content Exchange may also use the HTTP header to specify that some items are more important than other items, and should never be deleted from the cache, while other objects may only be pinned for short periods of time, or not at all. Objects may also be ranked in order of importance to indicate which ones should be deleted if garbage collection becomes necessary.

Among other things, Content Exchanges uses quotas to guarantee a certain amount of space to a content provider, report the amount of space used, detect over-runs and drop content when the level is exceeded. The quota is a combination of total space used and duration; as new content comes in or old content is removed, the proxy cache monitors the total amount of objects in the cache for that domain and compares it to the content. Any over-runs are detected and the appropriate steps taken to report the problem and delete content that is determined least essential.

## Feedback on new content

### Standard model

Until now web proxy cache feedback has been limited to log file aggregation over long periods of time. This is sufficient for billing and tracking of access patterns, but inadequate for rapid feedback and diagnosis of problems. Among other things, this may result in a content provider doubting the immediate benefits of caching and on operators being unable to fix problems.

### Content Exchange

Content Exchange addresses the feedback on new content problem by storing cache exchange requests as they are received. The requests remain in a queue until they are processed, and are moved to a retry queue when they fail. By examining the contents of the queue, Content Exchange can provide immediate feedback to a content provider on which caches and content have been updated and which have not.

## Distributed Log Entry Aggregation

### Standard model

In a traditional network of web proxy servers, access logs are generated to record each and every access through the proxy. These logs are then collected at collation points and processed using off-line analysis programs and scripts, resulting in a set of data that represents the aggregate activity of the proxies involved.

The problem of this traditional approach is that the amount of raw log data being generated by each proxy is very large. Consider that a proxy operating at a reasonable speed of 200 accesses/second with an average access log entry size of 100 bytes. Thus this proxy generates roughly 20KB of data per second, or almost 2GB of data per day. Now multiply this by the number of proxies sending their data to a collation node, and it becomes obvious that the amount of data to be processed daily is huge. Additionally, requiring the each proxy send almost 2GB of data per day over the network places an additional strain on an increasingly-congested network.

### Content Exchange

One solution might simply be to sample the data which at a reasonable sampling interval can produce statistically significant information about the state of a proxy. However, this solution will not work for Content Exchange because of the need to account for each access individually and pass that information back to the clients for billing information. In other words, once logs are used for billing information, complete and accurate reporting is a necessity.

For Content Exchange, we have developed a new solution to the log data problem called Log Aggregation. The concept of Log Aggregation is to collect information about each and every

access through a proxy, just as is currently being done, but rather than simply creating a log entry with all of the information, it is stored internally and aggregated locally with other entries before a single entry is written to the logfile. This aggregation is performed over a given, user-definable interval of seconds. For example, if the interval is defined to be one second and the proxy is generating 200 accesses per second, then after 5 seconds the traditional logfile will contain roughly 1000 entries whereas the aggregate logfile will contain just 5, where each entry represents aggregate counts for roughly 200 entries each.

In order to create an aggregate log entry, a user must define how the entry will appear using a set of aggregation operators. These operators are used to aggregate each successive access entry into the aggregate total, which is written out at each aggregation interval. The current set of aggregation operators are:

<b>SUM(field)</b>	Sum each entry within the interval
<b>COUNT(field)</b>	Count of the entries within the interval
<b>AVERAGE(field)</b>	Average of the entries within the interval
<b>FIRST(field)</b>	First entry seen in the interval
<b>LAST(field)</b>	Last entry seen in the interval

While this list represents the most popular aggregation operations, it is not intended to be complete, and this design can accomodate new operators as long as the operation can be defined for a set of values within an interval.

As an example, consider that we have a log file which records the time of access, number of bytes sent through the proxy, and the number of milliseconds required to process the request. Then a logfile of 10 such entries might appear as:

```
943053611 256 130
943053611 1024 200
943053611 256 120
943053611 512 140
943053611 128 100
943053612 256 200
943053612 2048 400
943053612 64 30
943053612 256 120
943053612 128 90
```

Now consider that we create an Aggregate Log with the following definition:

**LAST(access\_time) SUM(bytes\_served) AVERAGE(time\_to\_process)**

If this log were specified with a 1 second interval, then the result would be a logfile with the following two entries:

```
943053611 2176 138
943053612 2752 168
```

Another aspect of this feature is that log analysis can now be accomplished in a distributed, parallel manner, since each proxy node is computing ongoing aggregates at the same time. When these aggregate logs are then collated for centralized processing, the amount of data is reduced by several orders of magnitude, greatly simplifying processing time and reducing the strain on the network.

Implementation of this feature is done by collecting the fields that are being aggregated into accumulators and then processing the accumulators at the end of the interval. With two

counters and a user-defined post-processing step, all of the current aggregation operators can be supported, as well as many others. Another aspect of the implementation is a grammar for specifying the logfile format and interval time, such as in the example above. For our implementation, we have chosen to use SQL-like operators and a new XML-based configuration system that helped to define all types of logfile formats, even aggregates.

Content Exchange places a certain burden on all participating parties to collect and report accurate access information so that the billing model can proceed with confidence. This increased level of log reporting may overwhelm already busy networks and proxy servers if certain innovative measures are not taken. Our new distributed log aggregation feature provides a detailed set of information for billing, but at a fraction of the cost in terms of data that needs to be sent through the network and analyzed.

## **Billing and Statistics collection**

### **Standard proxy caches**

Because all data is treated the same, the only way a hoster or access provider makes money from proxy caches is by making the network more efficient without adding costly hardware, and perhaps by attracting and keeping customers. Billing is not generally an issue, but it is essential for the success of Content Exchange, which guarantees the availability of content in the cache.

Collecting access statistics from proxy caches is generally difficult for content providers because the origin server never processes the requests that result in a hit in the proxy cache. Content providers generally use invisible, non-cacheable zero-length content in their HTML pages to get access information from proxy caches. Since this results in a trip to the origin server merely to collect statistics, it slows the user's access of the content and uses network bandwidth.

### **Content Exchange**

Content Exchange uses distributed log entry aggregation to collect access statistics including:

- number of bytes served
- amount of storage reserved
- amount of storage used
- duration of the storage
- transfer rate compared to estimated transfer rate using origin server.

Any combination of these items can be used to implement billing.

In addition, the level of detail in the aggregation will be controlled, resulting in providing different levels of service to a content provider. By collecting cache hit information, the hoster or access provider can provide an accurate picture of which data is being accessed and how often. Click-through rates as well as patterns of click throughs can be generated. User access to the content and network bandwidth are not effected because the calculations do not require a trip to the origin server to collect the data.

merely to collect statistics, it slows the user's access of the content and uses network bandwidth. Neither of these is a problem with Content Exchange since the data collection and tabulation occurs at the hoster or access provider.

The access information can be posted on a web site or sent directly to the content provider as quickly as it is generated, giving an up-to-date picture of the content access as well as being used for periodic purposes for billing.

#### Billing for Compressed Content

##### Standard proxy caches

Inktomi's Traffic Server proxy cache can be configured to compress content as it is stored, resulting in a more efficient use of cache storage and network. An entry is added to the HTTP header and sent to the browser with the content so it can be uncompressed at the browser and displayed.

The use of compressed content presents a problem for Content Exchange billing: since data is compressed, the number of bytes transferred will be only a fraction of the uncompressed content size, resulting in a significant loss of revenue.

##### Content Exchange

In order to calculate the number of uncompressed bytes transferred, Content Exchange enhances the compression software to store the uncompressed content length in the HTTP header which is stored in the proxy cache. The uncompressed content length is identified by a unique HTTP header field followed by the number of bytes. When present, this value is used for billing instead of the physical number of bytes transferred; when absent, the physical number of bytes transferred is used. The value could also be stored in a database or separate file, but this is not necessary.

In actual use large transfers of content are often aborted; for example, users become impatient and type another URL into the browser. For this reason, using the uncompressed content length may unfairly charge a Content Exchange content provider for an entire data transfer when only a portion of the transfer occurred. To fix this problem, Content Exchange multiplies the uncompressed content length by the ratio of transfer length and content length after compression. The transfer and content lengths are standard parts of the HTTP header.

The entire formula for calculating the number of uncompressed bytes transferred is expressed in the following pseudo-code and should be used in billing:

```
if (uncompressed_content_length exists) // data was compressed
    uncompressed_bytes_transferred =
        number_bytes_transferred / content_length *
        uncompressed_content_length
else
    // data was not compressed
    uncompressed_bytes_transferred =
        number_bytes_transferred
```

The calculation does not provide an exact value for the number of uncompressed bytes transferred because the compression ratio varies in a single piece of content. In most cases, however, it

does provide a reasonable estimate, and can be expected to be very accurate for large numbers of aborted transfers.

60175555-011500